Paper

Developing a Secure Image Steganographic System Using TPVD Adaptive LSB Matching Revisited Algorithm for Maximizing the Embedding Rate

P. Mohan Kumar[a] and K. L. Shanmuganathan[b]

[a] CSE Department, Jeppiaar Engineering College, Chennai, India
[b] CSE Department, R.M.K. Engineering College, Chennai, India

Abstract—Steganography is the approach for hiding any secret message in a variety of multimedia carriers like images, audio or video files. Whenever we are hiding a data, it is very important to make it invisible, so that it could be protected. A number of steganographic algorithms have been proposed based on this property of a steganographic system. This paper concentrates on integrating Tri way pixel value differencing approach and LSB matching revisited. The secret data embedded in images were images, text and audio signals so far. The proposed scheme has also come with the executable file as secret data. Also, the experimentation results show that, the important properties of a steganographic system such as imperceptibility, capacity of the carrier image and also resistance against the various steganalytic tools have also been achieved with this stego-system.

Keywords—executable file, LSBMR, spatial domain, steganalysis, TPVD.

1. Introduction

The term steganography means the science of hidden communication. The way in which steganography differs from another secure data communication technique called cryptography is, the visibility of the data exchange. In cryptography, even though the actual data transaction may not be known to a third person, he may get a doubt that some abnormal or suspicious communication is taking place. But, in case of steganography, the hidden communication will never come to the notice of the eavesdropper. Because, the carrier signal we are using to hide the secret data is going to be innocent. So, we can call the technique as information hiding [1]–[4].

Another technique which is based on the information hiding strategy is digital watermarking. But, in case of digital watermarking, the important property of information hiding known as resistance to removal is preferred. So, in these applications, we are not worrying about imperceptibility but resistance to removal. This is mainly used in commercial applications like copyright protection of digital forms of media like video or image. Unlike image steganography, digital watermarking techniques mainly concentrate on keeping logos or any other symbols or images in the carrier data. And also it is made sure that those signals embedded are not able to be removed by any other person. There are a number of watermarking techniques have been explained in [5]–[7].

For a long period of time many researchers have been involved in developing new steganographic systems. Meanwhile, the development of steganalytic tools are also started growing. Steganalysis is a process of finding the existence of a secret data in a cover media [8]. Whenever a suspicious image is received, the main task of a steganalytic tool is to find the algorithm used for hiding secret data in the image. Most of the steganographic algorithm developers are also trying to crack their own algorithm using the existing steganalytic tools, so that the strength and weaknesses of their system may be found.

2. Related Work

Generally, digital image steganography is a way to exchange secret data. So, the important components of a steganographic system include an embedding/extracting algorithm, secret key which is going to be shared by the sender and receiver of the secret data and also a communication channel which is considered to be more secure [9]. The general frame work for a steganographic system is shown in Fig. 1.

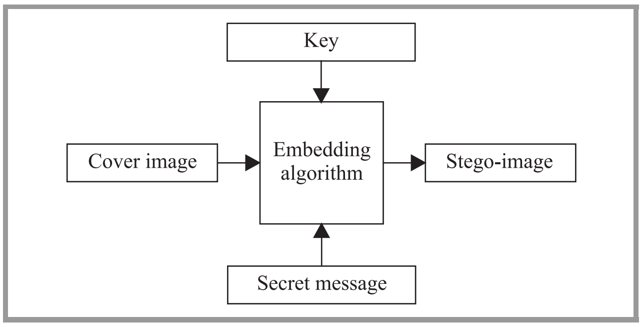

Fig. 1. A simple image steganography scheme.

This framework has been derived from the popular idea called prisoner's problem [9]. In this approach Alice and Bob were trying to exchange an escape plan without the knowledge of the warden. Some of the terms used in steganographic system are cover-media (the digital media which is used to hide secret data), secret data (the important data to be hidden) and stego-media (after embedding the secret message in the cover media). The hidden data cannot be detectable when we are performing the embedding phase randomly and also the level of independence between the secret message and cover as well as stego objects [10]. There are many other ways for providing more security includes the usage of encryption-decryption functions for embedding and extraction of secret data [11]. Since JPEG images are widely exchanged through internet, choosing JPEG image for sending secret message to the receiver will never be suspicious. And also, the redundancies that are appearing in JPEG images help us to hide more information securely. Methods for improving the hiding capacity of a JPEG image have been explained in [12].

The least significant bits replacement method or LSB method is the very simple and a commonly used approach for developing steganographic system. Because the amount of space that an image can provide for hiding data will be more comparing with other algorithms. And also the implementation of this technique is also very easy. In this approach, the image pixel's LSB is replaced by one bit of secret data [13]. Spatial domain embedding technique is also known as image domain. The techniques that are following spatial domain embedding are embedding the secret message in the intensity of the cover image pixels. Spatial domain techniques include bit-wise methods that apply bit insertion and noise manipulation techniques [14]. The main disadvantage of LSB replacement is that, while hiding secret data in the image, some of the pixels will never be modified or replaced with the secret bits, since we are using pseudo random generator for placing the secret message bits. As a result, very simple steganalytic tool could trace the existence of the secret message.

But this problem of asymmetry can easily be avoided by an alternate scheme using a LSB matching scheme. In this technique, if the secret bit is not matching with the LSB of cover image, then ±1 will be added randomly. By doing so, we can reduce the probability of increase or decrease in the pixel value modification can be avoided. So, we can eliminate the problem we faced in LSB replacement technique. Also, the steganalytic algorithms which can find the stego-images which were obtained from LSB replacement technique cannot find the stego-images we got from LSB matching.

There are several steganalytic algorithms found for finding stego-images which were got by LSBM (LSB matching) technique. In [15], the image is being taken and its two least significant bit planes are considered. The bit planes are split into 3×3 overlapped sub images. According to the number of gray levels those sub images are classified. In one sub image, the LSBM is applied and found that the alteration rate of cover image is higher than that of stego-image. In [16], the authors have compared the function of LSBM to a low pass filter through the histogram of the image. They found that the number of high frequency components is very less comparing to the original cover image. But later in [17], this method is found that it will not be working well in case of gray scale images. As a remedy, the author has proposed techniques using down-sampled image and adjacency histogram instead of traditional histogram.

Instead of handling pixel values independently, the other technique proposed by Jarno in [18], is using a pair of pixels for embedding which is known as LSBM revisited (LSBMR). In this technique, the author has proposed an approach for data hiding, in which the LSB of the first pixel carries one bit of information, and a function of the two pixel values carries another bit of information. So, in this approach the changes that are made in the cover image are very few. Also, the modification rates of pixels have been greatly reduced. But all the techniques analyzed above are not taking care of the relationship between the pixel and its neighborhood.

There are many data embedding schemes analyzed which are taking the relationship of a pixel to its neighbor. In [19], a hiding scheme has been proposed by replacing the LSB of a cover according to the difference values between a pixel and its four touching neighbors. This method uses the edges of an image for hiding secret data. Although this method can achieve more visually imperceptible stego-images, the security performance is poor. Since the method just modifies the LSB of image pixels when hiding data, it can be easily detected by existing steganalytic algorithms.

The pixel value differencing is another type of edge based data hiding scheme, which has been proposed in [20], in which the number of embedded bits is determined by the difference between a pixel and its neighbor. If there is large difference between the pixels, the number of secret bits that can be embedded will also be large. Also based on the experimental results, this approach can provide a larger embedding capacity.

Executable file structure

The program loader that is a subset of the Windows system assumes the loading executable files into a virtual memory, so the executable files have the format that the program loader can identify, and the format is called portable executable (PE). It is necessary to know the PE format and RVA which is an address type used in the PE in order to understand the new methods for hiding information in the PE. The system uses an image file as a cover to embed it to an executable program or an executable file for the proposed system.

The characteristics of the executable file does not have a standard size, like other files, for example the image file (BMP) the size of this file is between (2–10 MB). Other example is the text file (TEXT) the size often is less than 2 MB. Through the characteristics of files have been

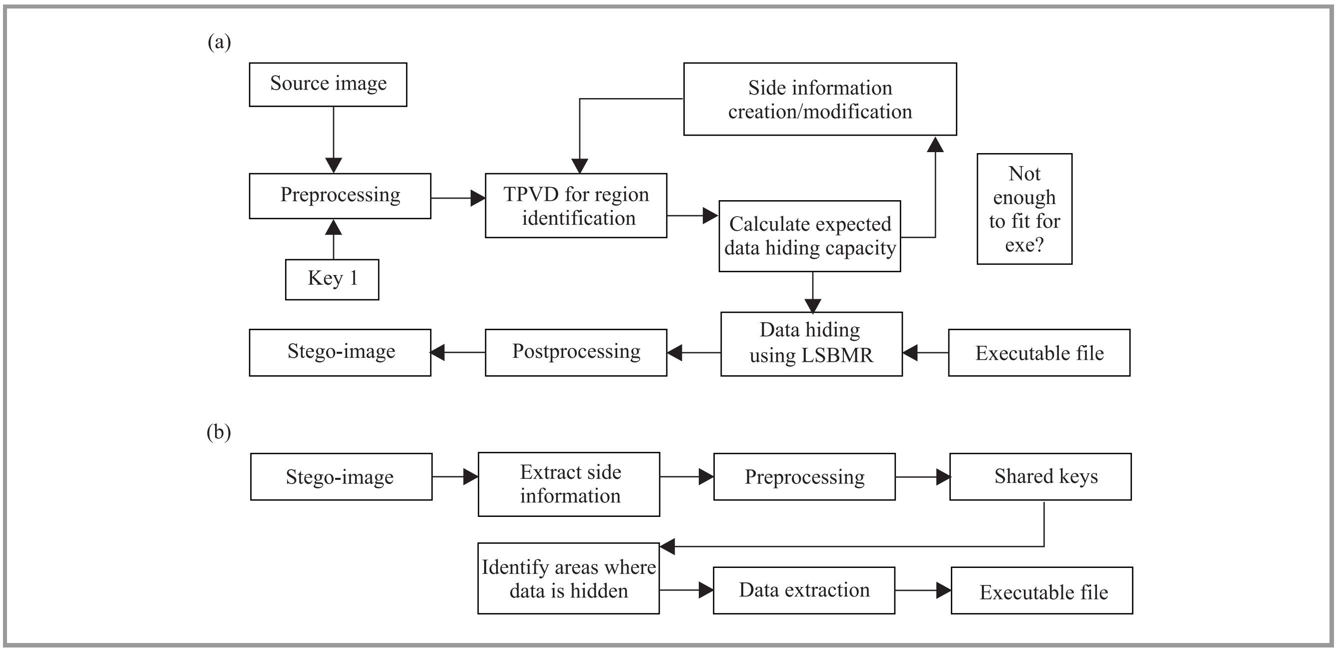

Fig. 2. Proposed executable file (a) embedding architecture and (b) extraction architecture.

used as a hidden information's, it found that lacks sufficient size to serve. For these features of the Executable file, it has unspecified size; it can be 650 MB like window setup file or 12 MB such as installation file of multi-media players. For taking advantage of this feature make it a suitable environment for concealing information without detect the file from attacker and discover the hidden information in stego-image.

A PE file section represents code or data of some sort. While code is just code, there are multiple types of data. Besides read/write program data (such as global variables), other types of data in sections include application program interface (API) import and export tables, resources, and relocations. Each section has its own set of in-memory attributes, including whether the section contains code, whether it's read-only or read/write, and whether the data in the section is shared between all processes using the executable file. Sections have two alignment values, one within the desk file and the other in memory. The PE file header specifies both of these values, which can differ. Each section starts at an offset that's some multiple of the alignment value.

3. Proposed System

The architecture for embedding phase of the proposed system is shown in Fig. 2a. The proposed system initializes some parameters, which are used for subsequent data preprocessing and region selection, and then estimates the capacity of those selected regions. If the regions are large enough for hiding the given secret message, then data hiding is performed on the selected regions. Finally, it does some postprocessing to obtain the stego-image. Otherwise the scheme needs to revise the parameters, and then repeats region selection and capacity estimation until can be embedded completely. Please note that the parameters may be different for different image content and secret message. We need them as side information to guarantee the validity of data extraction. In practice, such side information (7 bits in our work) can be embedded into a predetermined region of the image. In data extraction, the scheme first extracts the side information from the stego-image. Based on the side information, it then does some preprocessing and identifies the regions that have been used for data hiding. Finally, it obtains the secret message according to the corresponding extraction algorithm. In this paper, we apply such a region adaptive scheme to the spatial LSB domain. We use the absolute difference between two adjacent pixels as the criterion for region selection, and use LSBMR as the data hiding algorithm. The details of the data embedding and data extraction algorithms are as follows.

Data embedding

1. The cover image of size of m×n is first divided into non-overlapping blocks of 2×2 pixels. For each small block, we rotate it by a random degree in the range of $\{0, 90, 180, 270\}$, which is decided by a key which is decided by the user.

2. The resulting pixel blocks are $P(x, y)$, $P(x+1, y)$, $P(x, y+1)$ and $P(x+1, y+1)$. Consider the pixel $P(x,y)$ as the current pixel, consider the pixel pairs as $P1$, $P2$ and $P3$, where:

$$P1 = (P(x,y), P(x, y+1)),$$
$$P2 = (P(x,y), P(x+1, y)),$$
$$P3 = (P(x,y), P(x+1, y+1)).$$

3. Calculate the difference values for the pixel pairs keeping one pixel as the current pixel.

4. Find the appropriate range from the range table to identify the region or the pair of pixels (assume as x_i and x_{i+1} in which the secret data is going to be embedded.

5. We deal with the above embedding units in a pseudo-random order determined by a secret key. For each unit (x_i, x_{i+1}), we perform the data hiding according to the following four cases:

   (1) $\mathrm{LSB}(x_i) = m_i$ and $f(x_i, x_{i+1}) = m_{i+1}$ and $(x'_i, x'_{i+1}) = (x_i, x_{i+1})$,

   (2) $\mathrm{LSB}(x_i) = m_i$ and $f(x_i, x_{i+1}) \neq m_{i+1}$ and $(x'_i, x'_{i+1}) = (x_i, x_{i+1} + r)$,

   (3) $\mathrm{LSB}(x_i) \neq m_i$ and $f(x_{i-1}, x_{i+1}) = m_{i+1}$ and $(x'_i, x'_{i+1}) = (x_{i-1}, x_{i+1})$,

   (4) $\mathrm{LSB}(x_i) \neq m_i$ and $f(x_{i-1}, x_{i+1}) \neq m_{i+1}$ and $(x'_i, x'_{i+1}) = (x_{i+1}, x_{i+1})$,

   where m_i and m_{i+1} denote two secret bits to be embedded.

6. After data hiding, the resulting image is divided into non-overlapping 2×2 blocks. The blocks are then rotated by a random number of degrees based on key.

Data extraction

The architecture of the proposed system for executable file extraction is shown in Fig. 2b.

1. Partition the stego-image into 2×2 pixel blocks.

2. Calculate the difference values as we did in embedding phase.

3. Find the embedding location and then rotate by random degrees which is decided by the secret key.

4. Until all the hidden bits are extracted completely, go through all the pixel blocks whose difference is greater than or equal to the available cut-off value. This cut-off value will be the maximum value of the pixel that could be used for embedding data.

4. Experimental Results and Analysis

One of the important properties of our steganographic method is that it can first choose the sharper edge regions for data hiding according to the size of the secret message by adjusting a cut off value. As explained in the paper, the larger the number of secret bits to be embedded, the smaller the cut off becomes, which means that more embedding units with lower gradients in the cover image can be released. When is 0, all the embedding units within the cover become available. In such a case, our method can achieve the maximum embedding capacity of 100% (100% means 1 bpp on average for all the methods in this paper), and therefore, the embedding capacity of our proposed method is almost the same as the LSBM and LSBMR methods except for 7 additional bits. One of the sample cover image taken for the experimentation and the corresponding stego-image with executable file are shown in Fig. 3. There are no visual artifacts found in the stego-image so that the stego image is having the hidden exe file which could not be identified by the human visual system (HVS). A comparison of accuracy of RS features between the existing and proposed methods are provided in Table 1. In Fig. 4, the LSB planes of the cover image and

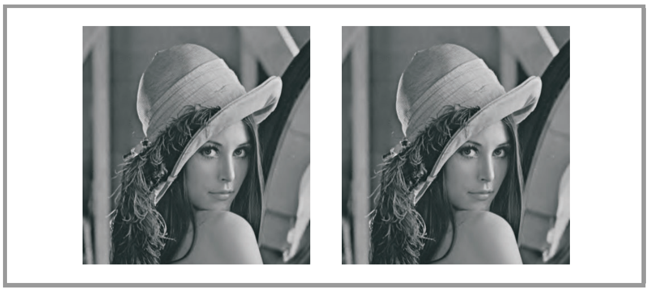

Fig. 3. Cover image (a) and (b) stego-image using proposed approach.

Table 1
Average accuracy [%] of RS features set on FLD

Embedding rate methods	10%	20%	30%	40%	50%
Existing	88	91	94	98	99
Proposed	51	52	51	50	53

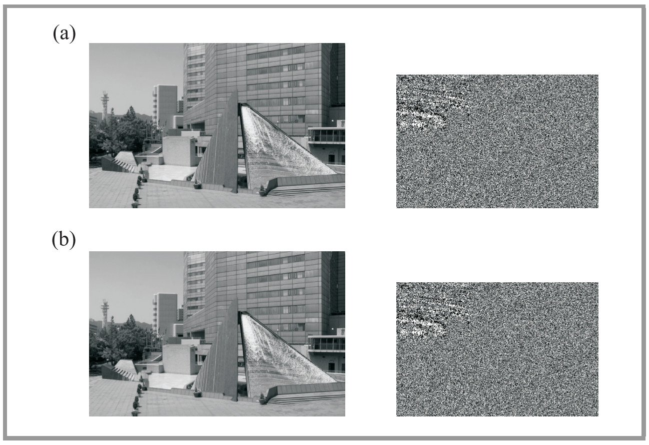

Fig. 4. The LSB planes of (a) the cover image and (b) the stego-image.

stego-image are given. Based on the histogram analysis the cover image could not be suspected so that this method is producing stego images which will not be traced by the existing steganalytic algorithms. Table 2 contains the data for drawing ROC curves and RS diagram for the proposed system in comparison with the existing system is shown in Fig. 5.

Table 2
Data for drawing ROC curves

False positive rate	0	0.1	0.2	0.3	0.5
True positive rate	0.3	0.5	0.6	0.65	0.7

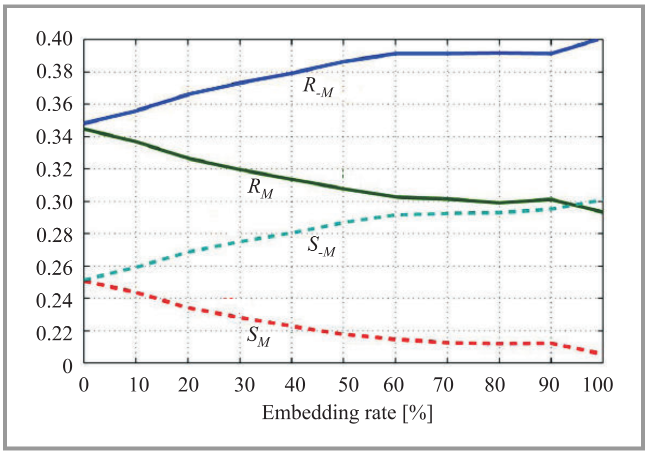

Fig. 5. RS diagram for proposed method.

5. Conclusion

In this paper, an image steganographic scheme in the spatial LSB domain is studied in which an edge based scheme also included. Normally, there exist some smooth regions in natural images, which would cause the LSB of cover images not to be completely random or even to contain some texture information just like those in higher bit planes. If embedding a message in these regions, the LSB of stego-images becomes more random, and according to our analysis and extensive experiments, it is easier to detect. In most previous steganographic schemes, however, the pixel/pixel-pair selection is mainly determined by a PRNG without considering the relationship between the characteristics of content regions and the size of the secret message to be embedded, which means that those smooth/flat regions will be also contaminated by such a random selection scheme even if there are many available edge regions with good hiding characteristics. The experimental results evaluated on thousands of natural images using different kinds of steganalytic algorithms show that both visual quality and security of our stego-images are improved significantly compared to typical LSB-based approaches and their edge adaptive versions. Furthermore, it is expected that our adaptive idea can be extended to other steganographic methods such as audio/video steganography in the spatial or frequency domains when the embedding rate is less than the maximal amount.

References

[1] N. Provos and P. Honeyman, "Hide and seek: an introduction to steganography", *IEEE Sec. Priv. Mag.*, vol. 1, no. 3, pp. 32–44, 2003.

[2] J. Fridrich, "Applications of data hiding in digital images", in *Proc. Int. Symp. Sign. Process. Apl.*, Brisbone, Australia, 1999, pp. 22–25.

[3] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding – a survey", *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[4] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding", *IBM Syst. J.*, vol. 35, no. 3–4, pp. 313–336, 1996.

[5] M. Swanson, M. Kobayashi, and A. Tewfik, "Multimedia data embedding and watermarking technologies", *IEEE Proc.*, vol. 86, no. 6, pp. 1064–1087, 1998.

[6] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia", in *Proc. First Int. Worksh. Inf. Hiding*, R. Anderson, Ed. Cambridge: Springer, 1996, pp. 183–206.

[7] I. J. Cox, M. L. Miller, and J.A . Bloom, *Digital Watermarking*. Morgan Kaufmann, 2002.

[8] T. Pevný and J. Fridrich, Multiclass detector of current steganographic methods for JPEG format, *IEEE Trans. Inf. Forensics Sec.*, vol. 3, no. 4, pp. 635–650, 2008.

[9] G. Simmons," The prisoner's problem and the subliminal channel, *CRYPTO*, pp. 51–67, 1983.

[10] J. Zollner and H. Federrath, "Modelling the security of steganographic systems", in *Proc. 2nd Inf. Hiding Worksh.*, Portland, USA, 1998, pp. 345–355.

[11] N. J. Hopper, J. Langford, and L. Von Ahn, "Provably secure steganography", in *Advances in Cryptology: CRYPTO 2002*. Springer, 2002.

[12] L. Zhang, H. Wang, and R. Wu, "A high capacity steganography scheme for JPEG 2000 baseline system", *IEEE Trans. Image Process.*, vol. 18, no. 8, 2009.

[13] T. Morkel, J. H. P. Eloff, and M. S. Olivier, "An overview of image steganography" [Online]. Available: http://mo.co.za/open/stegoverview.pdf

[14] N.F. Johnson and S. Jajodia, "Steganalysis of images created using current steganography software", in *Proc. 2nd Inf. Hiding Worksh.*, Portland, USA, 1998.

[15] F. Huang, B. Li, and J. Huang, "Attack LSB matching steganography by counting alteration rate of the number of neighbourhood gray levels", in *Proc. IEEE Int. Conf. Image Process.*, San Antonio, USA, 2007, vol. 1, pp. 401–404.

[16] J. Harmsen and W. Pearlman, "Steganalysis of additive-noise modelable information hiding", in *Proc. SPIE Electronic Imaging*, Santa Clara, USA, 2003, vol. 5020, pp. 131–142.

[17] A. D. Ker, "Steganalysis of LSB matching in grayscale images", *IEEE Sig. Process. Lett.*, vol. 12, no. 6, pp. 441–444, 2005.

[18] J. Mielikainen, "LSB matching revisited", *IEEE Sig. Process. Lett.*, vol. 13, no. 5, pp. 285–287, 2006.

[19] K. Hempstalk, "Hiding behind corners: using edges in images for better steganography", in *Proc. Comput. Women's Congress*, Hamilton, New Zealand, 2006.

[20] D. Wu and W. Tsai, "A steganographic method for images by pixel-value differencing", *Pattern Recognit. Lett.*, vol. 24, pp. 1613–1626, 2003.

P. Mohan Kumar, B.E., M.E., Ph.D., works as Associate Professor in Jeppiaar Engineering College and he has more than 8 years of teaching experience. His areas of specializations are Network security, Image processing and artificial intelligence.

e-mail: mohankumarmohan@gmail.com
Jeppiaar Engineering College
Chennai, India

K. L. Shanmuganathan, B.E, M.E., M.Sc., Ph.D., works as the Professor and Head of CSE Department of RMK Engineering College, Chennai, TamilNadu, India. He has more than 18 years of teaching experience and his areas of specializations are Artificial Intelligence, Computer Networks and DBMS.

e-mail: kls_nathan@yahoo.com
R.M.K. Engineering College
Chennai, India